Briana Long

8/24/2025

CS 470 Final Reflection

https://www.youtube.com/watch?v=Jm7jPyhihN0&ab_channel=Brianal

Experiences and Strengths:

Explain how this course will help you in reaching your professional goals.

This course has helped me achieve personal learning goals, such as using Amazon Web Services. I plan on continuing learning how to use AWS to achieve a certification. I have learned how to build and deploy a full-stack web application in a cloud-based environment. This is a necessary skill in professional environments. By adding this to my portfolio, I will be able to meet job posting requirements.

What skills have you learned, developed, or mastered in this course to help you become a more marketable candidate in your career field?

This course has helped me develop my skills in containerization. With Docker, I was able to manage multicontainer deployments, which is something I have not done yet. I learned how to implement a RESTful API to connect the front and backend of the project. This was my first time working with AWS to host, scale, secure, and test my application. This required me to troubleshoot with testing, as I had to redo the project fully once.

Describe your strengths as a software developer.

My strengths as a software developer include problem-solving skills and communication. I found breaking down the steps and creating a checklist were incredibly helpful. I ran into issues. I was able to go a step and repeat the instructions to get the desired results. My communication skills are a strength, as I am able to communicate technical concepts to people in nontechnical roles.

Identify the types of roles you are prepared to assume in a new job.

- Full Stack Developer
- Cloud Developer
- DevOps Engineer
- Software Engineer

Planning for Growth:

Synthesize the knowledge you have gathered about cloud services.

Learning how to set up and scale applications is an important skill when working with cloud services. This is a necessary skill as planning for growth ensures that the application and cloud can handle an increase in users, larger workloads, and maintain a cost-effective efficiency.

Identify various ways that microservices or serverless may be used to produce efficiencies of management and scale in your web application in the future. Consider the following:

- How would you handle scale and error handling?
- How would you predict the cost?
- What is more cost-predictable, containers or serverless?

Microservices break the application down into smaller, independent services. Services can include authentication, payments, and analytics of the system. This creates an easier management system that can be scaled on demand. Serverless functions can be used for event tasks such as uploading, notifications, or background jobs. This reduces costs as the servers don't need to be continuously running.

Scale would be handled by using auto scaling groups, which would allow the flexibility needed without the extra charges. Load balancing is another way to distribute traffic. Error handling would be done through a monitoring system such as CloudWatch. Tools like this would allow monitoring and logging of activities.

Using a serverless architecture is the most cost-effective and predictive way to run a server. By using containers, there's a cost model for the workload currently running, since the charge is for using compute services. Serverless architecture is cost-efficient due to having low traffic options.

Explain several pros and cons that would be deciding factors in plans for expansion.

Containers:

Pros: Consistent environments, easy to migrate between clouds.

Cons: More management, pay for uptime regardless of usage.

Serverless:

Pros: No server management, instant scaling, pay per use.

Cons: Latency, vendor lock-in, and less environmental control.

What roles do elasticity and pay-for-service play in decision-making for planned future growth?

Cloud-based environments have the advantage of elasticity and pay for services. This makes it a great option compared to serverless architecture. Elasticity allows for auto-scaling of the services. This allows it to expand or shrink as needed to prevent larger costs. It's a great option for startups and small applications.